Project Documentation

Interactive Learning with HistoGlobe

Winter Semester 2014/15

Virtual Reality Group
Faculty of Media
Bauhaus University Weimar

Authors:
Marcus Kossatz
Chris Hornischer
Max Weber
Sebastian Köpsel

Supervisor :
Dipl.-Med.sys.wiss. Patrick Riehmann
M.Sc. Henning Gründl
Prof. Dr. Bernd Fröhlich

HistoGlobe®

VR-Systems Group
Bauhaus-Universität Weimar

Contents

1 Introduction

HistoGlobe is a interactive framework for people who want to learn history on another way than school.

2 User Centered Design Approach

HistoGlobe wanted to be helpful and usable for a specific target group: history teachers in schools. Therefore we looked for a school in the area of Weimar that could serve as a location for a field study. We found a school in Jena, 25 km east of Weimar, that offered us to develop an instance of HistoGlobe directly for the usage in class.

Lobdeburgschule in Jena-Lobeda is a public school for all students from grade 1 to 13. A history teacher in grade 12 invited us to conduct a field study in his class to test HistoGlobe directly in school. This gave us the chance to develop the visualization in a User Centered Design approach thoughout the semester. Two members of the project group went every two or three weeks to the teacher in Jena in the time from October 2014 until April 2015. We presented new concepts, asked specific questions about the interface and the usage of the visualization in class and new problems and questions about the concept raised that had to be clarified until the next meeting.

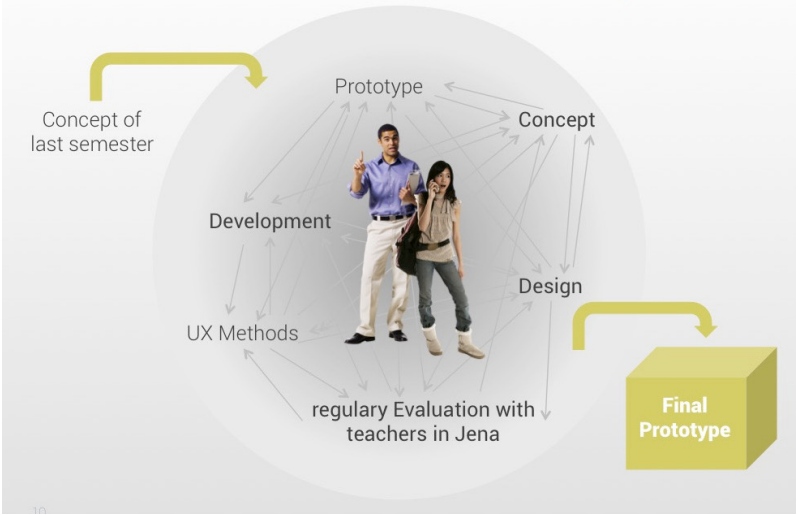


Figure 2.1: User Centered Design

Design Iterations

We had a lot of different design concepts. On the one hand we wanted to maximize the utility for the teacher to help him convey the necessary information in class but on the other hand design HistoGlobe in a way that we found suitable. We played around with orientation and the functionality of the timeline, the information about historical events on the map or the colors of the interface.

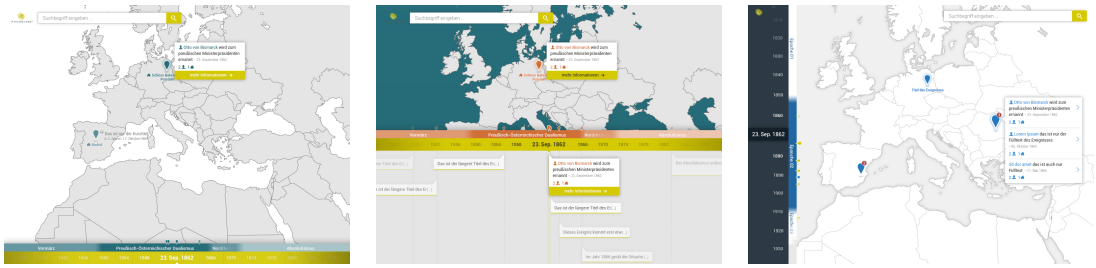


Figure 2.2: Several iterations of the design throughout the semester

In the next chapter we want to introduce the final elements of the user interface that were the result of the design iterations with the teacher.

3 User Interface Elements

The interface consists of five main elements: The **Map** is the central element showing the current countries with their names and their borders and the position of historical events, calles **Hivents** happening around the current date. This date is set on the **Timeline** which allows to control the temporal dimension: Set a new date and see the status of this day in history on the map. There are also **Topic Bars** on top of the timeline showing historical epochs in a specific time period. The sidebar on the right contains a **Search Bar** for retrieving information about historical events and a **Hivent List** for hivents of the selected topic. If an hivent is selected, there is a **Hivent Box** opening presenting the name, a short description and an image or video about the Hivent. Additionally there are **Control Buttons** for zooming the map or timeline, toggling the full screen or high contrast modus for better readability in problematic lighting conditions in the classroom.

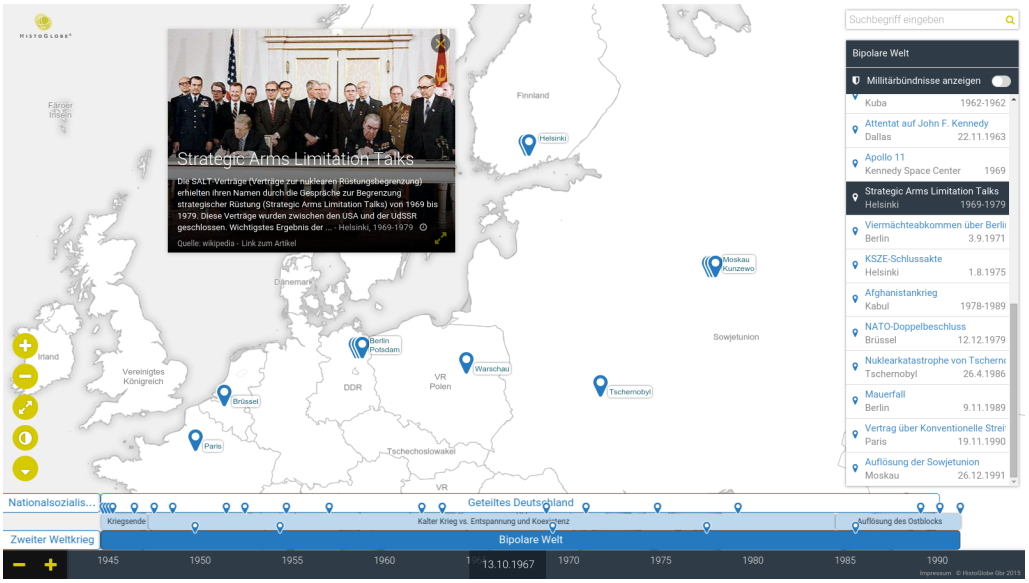


Figure 3.1: The final User Interface of HistoGlobe

3.1 Map

The map shows the status of the countries on Earth at this certain moment of history set in the timeline – the **NowDate**. For this project we used a self-made dataset of historic countries of whole Europe from 1945 until today and from Western, Northern, Southern and Central Europe from 1871 until 1945. We organized the data in a way that we can visualize historic changes of countries on the map. Finally we provided a functionality to style the areas due to a current theme, for example all countries belonging to NATO at the NowDate get a blue background color.

3.1.1 Historic Countries

A country consists of an **area**, represented as a multipolygon geometry and a **label** with the name of the country and the position of the label.

Areas Everything is based on a dataset of the current countries in Europe from *Natural Earth Data* [1]. We extracted only the countries of Europe and loaded them into *QuantumGIS*, an open

[1] 1:10m Cultural Vectors | `http://www.naturalearthdata.com/downloads/10m-cultural-vectors/`

source GIS software for organizing, analyzing and visualizing areas on Earth. For each historic country we found an historic map online and created the area of the country using the *Vector Geoprocessing Tools* of QuantumGIS. Each area is stored in a single `area_id.geojson` file.

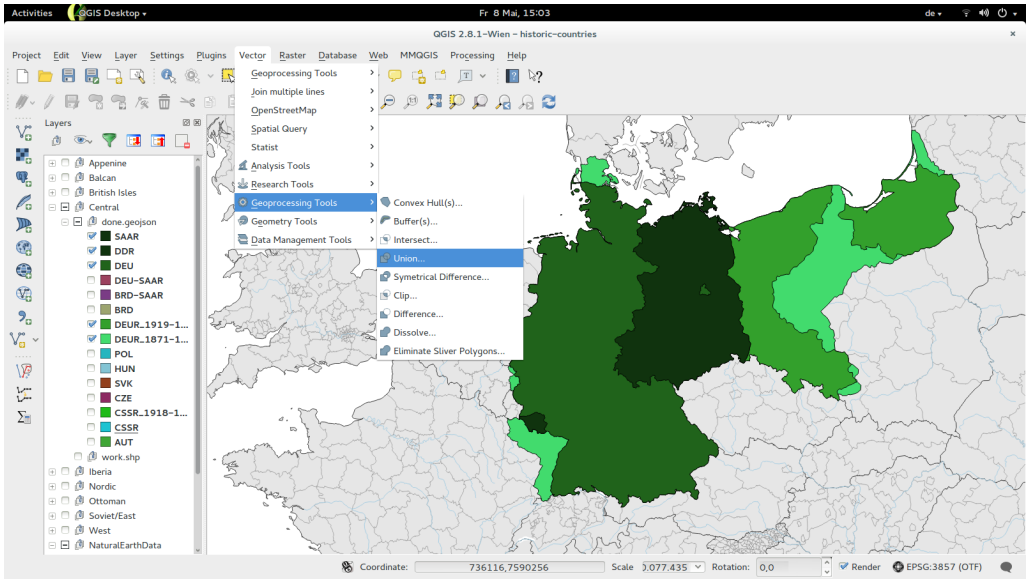


Figure 3.2: Geometry Manipulation of Historic Countries in QuantumGIS

In 3.2 you can see the areas of Germany from 1871 until today, from light green to dark green:

- 1871 - 1919 German Empire
- 1919 - 1945 Weimar Republic and Third German Reich (after WW I)
- 1945 - 1949 Occupied Germany (after WW II)
- 1949 - 1990 GDR without West Berlin and
  1949 - 1956 Saarland

Because of the very problematic Usability of QuantumGIS and the mass of data that would have needed to be processed we have not reached the goal to create a data base of all historic countries in Europe from 1871 on, but only from 1945 on, due to the time constraint.

Labels of a country are set in an table consiting of the id, the label name, the label position and a priority of a label. Labels are stored separately from the areas to account for independent changes of names and geometries: A country can be renamed and borders can change, but both events do not need to correlate. The list of labels is stored in a table `labels.csv`.
The problem with this approach is that the label position and priority can not be deducted from the area. Therefore, both have to be set by hand. We are aware that this manual approach is not optimal, but for the scope of that project it is suitable. For the future, a data model should be found in which areas and labels are connected but can still change separately from each other.

3.1.2 Historic Changes

Because of the way areas and labels are organized, an historic change can easily be modelled:

Table 3.1: Examples of Historic Changes

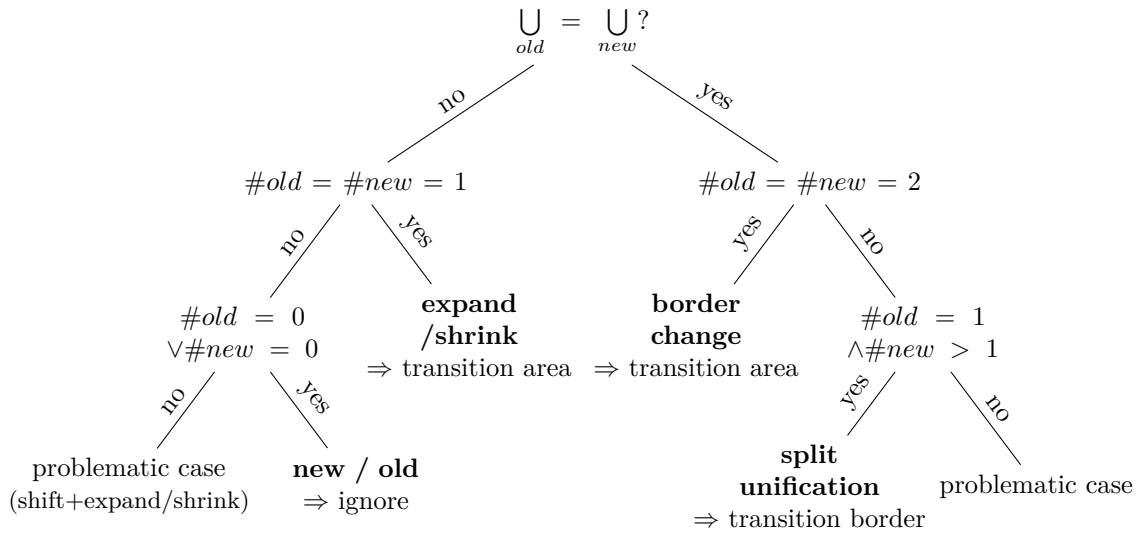
date	**name of change**	
domain	**old**	**new**
25.12.1991	Dissolution of Soviet Union	
area:	CCCP	EST, LVA, LTU, BLR, UKR, MDA, RUS
label:	CCCP	EST, LVA, LTU, BLR, UKR, MDA, RUS
03.10.1990	German reunification	
area:	BRD, DDR	DEU
label:	BRD, DDR	DEU
01.01.1990	End of Socialistic Republics	
area:		
label:	PR-ROU, PR-BGR, PR-HUN, PR-POL	ROU, BGR, HUN, POL
01.01.1979	Separation of Greenland	
area:	DNK-with-GRL	DNK, GRL
label:		GRL
01.01.1881	Init state	
area:		DEU-REICH-1871, POL-1871, ...
label:		DEU-REICH-1871, POL-1871, ...

This event-based data model is maintaines like this: if an historic change appears, it needs to have a date of change, a description of what happened, a set of areas that stop exist and a set that starts to exist from this date in history on – the same for the labels. Afterwards, the new areas have to be created in *QuantumGIS*, which is most of the work, and new labels have to be defined in the table.

For the future, an editor for storing, managing and analyzing historic changes would be desireable, because the data aqcuisiton part took a large share of the projects time.

In order to visualize the data on the client side, the areas, labels and changes have to be preprocessed on the server. Especially the transition areas and borders have to be generated.

Transitions are the geometric changes in a change event. There is either a transition area, which is the area that changes the membership of a country (e.g. Alsace-Lorraine 1919 from the German Empire to France) or a transition border, that splits two countries (e.g. The border between Czech and Slovak Republic after the dissolution of Czechoslowakia in 1991). These transitions shall be emphasized with an animation in the moment of the historic change so that it is clearly visible to the user what is currently happening. The transitions are generated like this: For each historic change the set of old and new areas are compared to each other and passes the following decision tree:



Legend: *old* = set of all old areas, *new* = set of all new areas
$\bigcup_{old}$ = union of all old areas, #*old* = number of old areas

Preprocessing happens with a *Python* script performing the following steps:

1. loading the areas (from *geojson*), the labels and the changes (from *csv*)
2. checking the set of areas and labels for completeness and the changes for consistency
3. generating the transition areas
4. writing the data to *json* files to be delivered to the client:
   - 4.1 `areas.geojson`
   - 4.2 `labels.geojson`
   - 4.3 `trans_areas.geojson`
   - 4.4 `changes.json`

The Workflow at runtime of the program can be seen in 3.3. The diagram is simplified focussing only on the areas, but the process is the same for the labels.

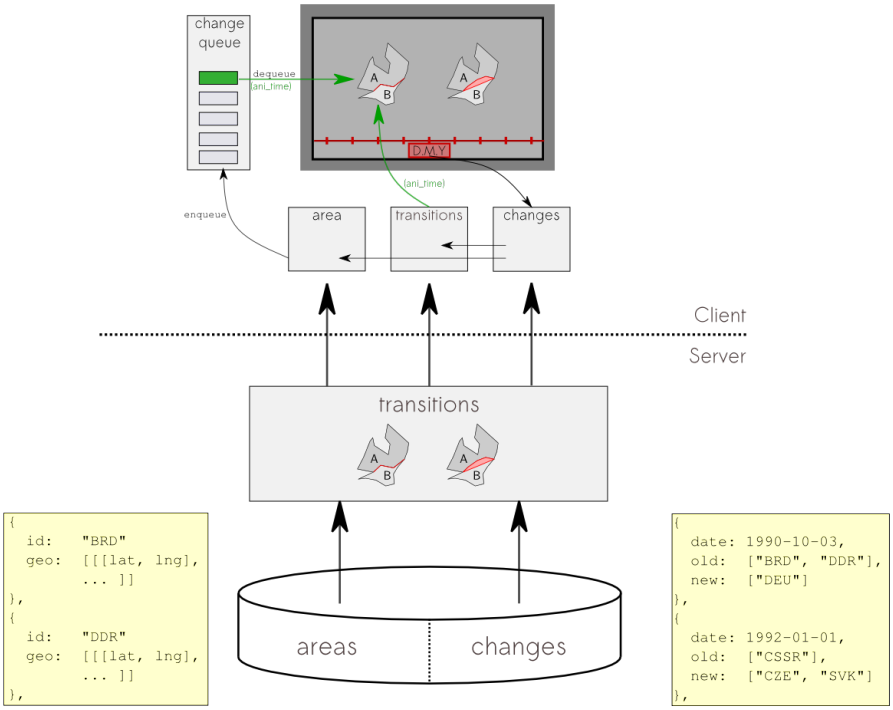


Figure 3.3: Architecture of historic countries on the map

The countries areas, labels and changes are created, preprocessed with the *Python* script and delivered to the server. The client gets the four *json* files and reads the data from there. When the NowDate on the timeline changes, the timeline sends the old date and the new date to Controller that finds out all the changes happened in time period. From each change the transition areas and borders are faded in on the map and the related old and new areas and labels are enqueued as a change event in a **change queue**. Every 50 milliseconds the queue is processed: for the first change event it checks if the related transitions are fully faded in. If so, the new areas and labels will be added and the old areas and labels deleted from the map. Finally, the transitions will be faded out again.

For moving the timeline backwards, the mechanism is the same, it is just that old and new areas and labels are swapped, because the historic change happens now the other way.

In order to prevent large amounts of changes on the map if the timeline is moved far, a rule-out mechanism is implemented: There is a list of old and new areas and labels for all historic changes in the period between the old and the new date from the timeline. Areas and labels that would be added in one change but deleted in another change are removed from both lists, because they would not contribute to the visualization. With this mechanism it is possible to move the timeline at a high speed there and back and always get a consistent update on the map without irritating the user.

3.1.3 Styling the Countries

3.2 Timeline and Topic Bars

Timeline von Mädelsmentor

3.3 Search Bar

Platzhalter

3.4 Hivent List

The central modul to navgate HistoGlobe in Histoglobe is the Hivent List. Here you can see Hivents with additional informations in one list ordered by date. Each Hivent has a name, a date and a location.

3.5 Hivent Boxes

Platzhalter

3.6 Control Buttons

Platzhalter

4 Field Study

Platzhalter

4.1 Usage as Teaching Material

4.2 Usage as Study Material

5 Conclusion

Platzhalter

5.1 Study Results

5.2 Discussion

5.3 Future Work